VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Implementace uživatelsky konfigurovatelných generátorů zátěže do systému Virtlab

Implementation of User-configurable Traffic Generators into Virtlab System

2010 **Bc. Ľubomír Prda**

Zadanie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne, uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave 7. mája 2010 .....

Moje poďakovanie patrí predovšetkým Ing. Petrovi Grygárkovi Ph.D. za odbornú pomoc a vedenie pri práci na diplomovej práci.

Abstrakt

Táto diplomová práca je venovaná problematike implementácie generátoru rôznych druhov tokov do systému Virtlab. Zaoberá sa výberov vhodných aplikácii pre generovanie tokov na sieti a návrhom užívateľsky konfigurovateľného systému pre ovládanie generátoru.

Kľúčové slová

Virtlab, generátor toku, tok v sieti

Abstract

Subject of diploma thesis is implementation of traffic generator for Virtlab System, capable of generating multiple types of traffic. Selection of suitable applications for traffic generation is considered, together with designing of an User-configurable system for traffic generator control.

Keywords

Virtlab, traffic generator, network traffic

Zoznam použitých symbolov a skratiek

XML	- Extensible Markup Language
OS	- Operačný system
IP	- Internet Protocol
TCP	- Transmission Control Protocol
UDP	- User Datagram Protocol
FTP	- File Transfer Protocol
VoIP	- Voice over Internet Protocol
DCCP	- Datagram Congestion Control Protocol
SCTP	- Stream Control Transmission Protocol
DNS	- Domain Name System
RTP	- Real-time Transport Protocol
CRTP	- Compressed Real-time Protocol
SIP	- Session Initiation Protocol
HTTP	- Hypertext Transfer Protocol
HTTPS	- Hypertext Transfer Protocol Secure
SSH	- Secure Shell
SCP	- Secure Copy
ICMP	- Internet Control Message Protocol
ISP	- Internet service provider
SRBD	- Systém Riadenia Bázy Dát
ER(D)	- Entity Relationship (Diagram)
DFD	- Data flow diagram

Obsah

Úvod

Počítačová sieť, kedysi výsada univerzít a vojenských projektov pre zdieľanie nazbieraných dát, dnes už neodmysliteľná súčasť výbavy každého počítača. Siete sa vyvíjali spolu s potrebami na ne. V minulosti postačovalo aby boli schopné preniesť dáta v presne definovanom formáte medzi bodmi A a B. Dnes spolu s rozmachom Internetu musia byť schopné preniesť dáta od textových súborov, cez multimédiá, až po interaktívne služby, telefonické hovory, hry a mnoho iného, medzi neobmedzený počet užívateľov zo všetkých kútov sveta. S rastúcimi možnosťami však rastie i zložitosť. Tak isto rastú i požiadavky na počítačové siete. V minulosti predstavovalo zvýšenie kvality iba zvýšenie prenosových rýchlostí na vyťaženom úseku. Pre využívanie novodobých služieb je však prosté zvýšenie rýchlosti nedostatočné. Kameňom úrazu sa stal práve rozdiel prenosov prostých dát, ktoré si zakladajú na prenosovej rýchlosti a prenosom multimediálnych dát a dát interaktívnych aplikácií, ktoré si zakladajú na rýchlom prenose menšieho množstva dát, s čo možnou najnižšou odozvou.

Pre potreby novodobých sietí boli vymyslené rôzne mechanizmy zabezpečujúce kvalitu služby. Konfigurácia takýchto systémov však nie je vôbec jednoduchá, pretože si vyžaduje preemptívne pochopenie dátových tokov na sieti. Pri rozmeroch sietí, akou je Internet, je nemožné predpokladať správanie sa všetkých tokov. Aj po dôkladnej analýze sa z konfigurácie stáva metóda pokusu a omylu. Pri nesprávnom zásahu však môže mať konfigurácia katastrofálne následky.

Pri školení odborníkov je potrebné, aby mal každý zo školených možnosť vidieť a pochopiť, ako jeho konfigurácia ovplyvní celkový stav počítačovej siete. Keďže väčšina nemá možnosť vyskúšať konfiguráciu na reálnej sieti, vznikajú rôzne systémy, simulujúce počítačové siete. Jedným z takýchto je i systém Virtlab.

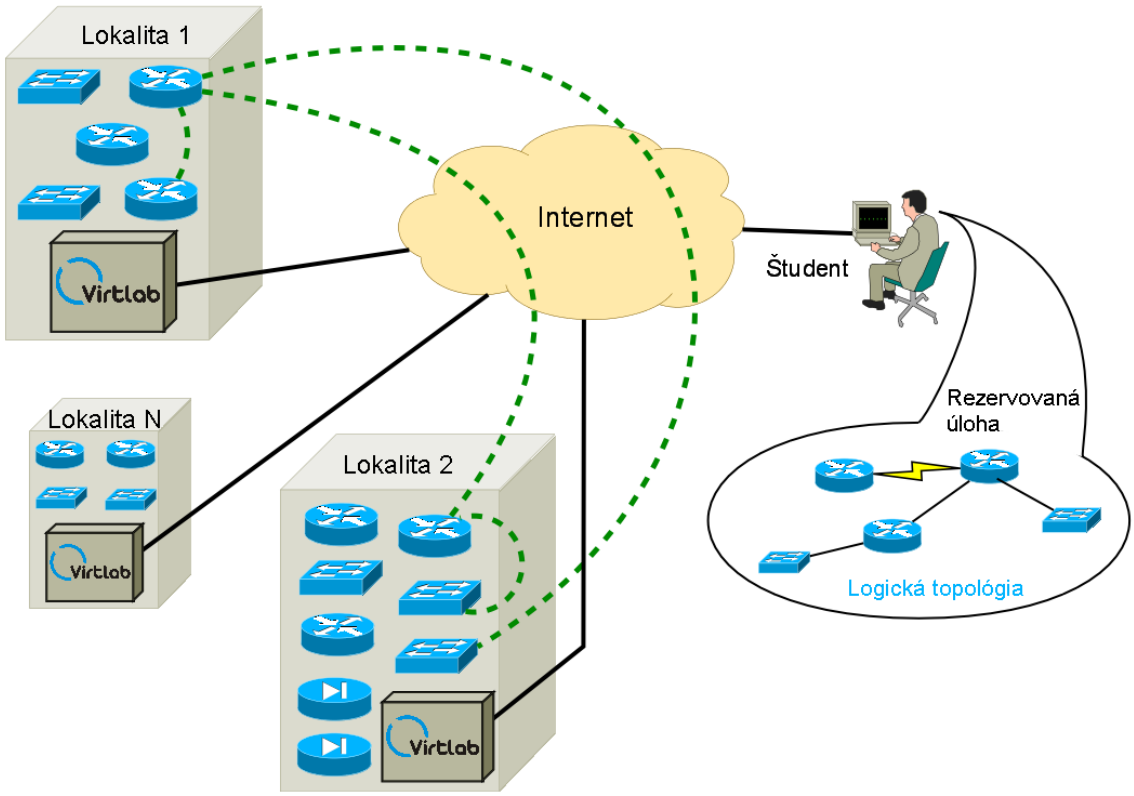
Cieľom tejto práce je vysvetliť čitateľovi základný princíp funkcie systému Virtlab. Práca sa ďalej zaoberá návrhom a implementáciou generátoru rôznych druhov dátových tokov pre tento systém. Dátové toky pritom musia čo najvernejšie kopírovať skutočné toky v počítačovej sieti. Práca pojednáva i o využití dostupných existujúcich generátorov a reálnych aplikácií, ktoré vytvárajú požadované toky a je ich možné zaintegrovať do systému.

Následne práca zjednocuje všetky získané atribúty do jedného užívateľsky konfigurovateľného systému. Systém má byť schopný konfigurovať, spúšťať a plánovať toky v sieti.

Záver práce pojednáva o možnostiach sledovania a vyhodnocovania generovaných tokov v systéme Virtlab.

1 Virtlab

Základnou ideou systému Virtlab je sprístupnenie laboratórnych prvkov pre prácu s počítačovými sieťami študentom na diaľku prostredníctvom Internetu. Študent si môže cez webové rozhranie zarezervovať laboratórne prvky na určitý čas a pristupovať k nim prostredníctvom webového prehliadača s podporou Java appletov. Rezervované prvky sú systémom prepojené podľa zvolenej úlohy, prípadne si môže študent zadefinovať vlastnú topológiu, ktorú mu systém pospája virtuálnymi linkami. Systém nie je viazaný používaním iba prvkov vo svojej lokalite, ale môže si „požičať“ prvok i z inej partnerskej lokality, na ktorej beží Virtlab. Vytvorenie topológie je ilustrované na obrázku 1.



Obrázok 1

Systém Virtlab sa skladá z nasledujúcich komponentov:

- **Riadiaci server lokality** – predstavuje komunikačné rozhranie medzi užívateľom a zvyškom systému. Je zodpovedný za mapovací proces, kedy sa mapujú prvky logickej topológie na skutočné zariadenia a generujú sa konfigurácie pre spojovacie zariadenia.
- **Rezervačný server lokality** – poskytuje popisy prvkov vo svojej lokalite a spracúva rezervačné požiadavky na prvky v svojej lokalite.

- **Konfiguračný server** – zabezpečuje príjem konfigurácií pre spojovacie zariadenie vo svojej lokalite.
- **Mazací server** – po ukončení rezervácie uvádza prvky do pôvodného stavu.
- **Konzolový server** – sprostredkováva komunikáciu medzi prvkami sieťovej topológie a zvyškom systému. Prijíma a autorizuje spojenia od klientov a premosťuje ich na porty pripojené k fyzickým konzolám laboratórnych prvkov. Komunikuje s konzolovými servermi v ostatných lokalitách.
- **Tunelovací server** – zabezpečuje vytváranie virtuálnych liniek medzi rôznymi lokalitami.
- **CRON** – stará sa o spúšťanie aktivačných a deaktivačných skriptov na začiatku a konci rezervovaného timeslotu.
- **Klientský applet** – emuluje terminál na strane klienta. Pripája sa na konzolový server.
- **Virtuálne distribuované spojovacie pole** – spojovací systém založený na prepojovaní pomocou VLAN (resp. QinQ) v rámci lokality a tunelovacím serverom medzi lokalitami.

V nasledujúcej časti práce bude podrobnejšie rozobraný konzolový server, ďalej bude popísaný mechanizmus rezervácie topológie a možnosti laboratórnych prvkov. Podrobnejšia znalosť ostatných komponentov Virtlabu nie je pre pochopenie systému navrhovaného touto prácou nutná.

1.1 Rezervácia topológie

Žiadosť o rezerváciu topológie dáva užívateľ riadiacemu serveru svojej lokality. Ten následnou komunikáciou s rezervačnými servermi lokalít zistí, či sú voľné prvky s požadovanými parametrami. Ak áno, prvky zarezervuje a namapuje na prvky logickej sieťovej topológie. Požiadavka na rezerváciu topológie môže prísť buď v podobe rezervácie preddefinovanej úlohy, alebo ako požiadavka na topológiu podľa priania užívateľa. V oboch prípadoch je logická topológia popísaná XML súborom nasledujúceho tvaru:

```
<!DOCTYPE virtual_topology SYSTEM "topology.dtd">
<virtual_topology>
      <edge technology="" name="" ether_type ="">
            <vertex name=""/>
            <vertex name=""/>
            <min_bps>
            </min_bps>
            <edge_feature>
            </edge_feature>
      </edge>
      ...
      <vertex_detail type="" name="">
            <vertex_feature>
            </vertex_feature>
            <os>
```

```
            </os>
        </vertex_detail>
        ...
</virtual_topology>
```

Výpis 1

Edge predstavuje prepojenia medzi prvkami topológie, charakterizované tagom vertex_detail. Pri prepojení „edge“ definujeme atribúty technology – typ prepojenia (serial/ehternet), name – pomenovanie prepojenia, ether_type – v prípade typu ethernet definuje typ ethernetu (legacy/fast/gigabit). A dva prvky vertex s atribútom name – názvy prepojovaných prvkov, ďalej min_bps – minimálnu požadovanú rýchosť spojenia a edge_feature – ďaľšie doplňujúce požiadavky.

Prvky topológie „**vertex“** majú definované atribúty type – typ prvku topológe (router/pc/switch), name – pomenovanie prvku namapovaného na reálny prvok. A ďalej prvky vertex_feature – dodatočné požiadavky na prvok a os – typ a verzia operačného systému prvku.

V oboch prípadoch neuvedením atribútov a prvkov ether_type, os, vertex_feature, edge_feature a min_bps, ktoré nie sú povinné, špecifikujeme, že na daných vlastnostiach nezáleží.

1.2 Laboratórne prvky

V súčasnej dobe sú súčasťou laboratórnych prvkov routre a switche spoločnosti Cisco s operačným systémom IOS zvládajúcim základný routing a switching. Novšie zariadenia obsahujú aj obrazy systému s ďalšími podpornými aplikáciami ako sú analyzátory a generátory tokov. Tieto zariadenia je možné zarezervovať s odpovedajúcim atribútom OS v XML pre registráciu topológie.

Ďalej sú súčasťou virtuálne pc s operačným systémom Linux, bežiacich na virtuálnej platforme XEN. Systém je schopný bežať obecne z ľubovoľným operačným systémom, ktorý je konfigurovateľný cez konzolu. Vhodnosť tej ktorej virtuálne platformy pre beh je zachytaný v diplomovej práci Bc. Pavla Jušku. Ďalšie požiadavky pre systém nie sú dané. Zariadenia obsahujúce špecifické softwarové vybavenie je opäť možné zarezervovať s odpovedajúcim atribútom OS v XML pre registráciu topológie.

1.3 Konzolový server

Jeho úlohou je sprostredkovať prístup k sériovým a telnetovým konzolám jednotlivých zariadení, prostredníctvom jednoduchého protokolu nad TCP/IP. Využíva ho Java Applet bežiaci vo webovom rozhraní, čo umožňuje užívateľom prístup ku konzolám zariadení. Dokáže sprostredkovať spojenie s konzolovým serverom inej lokality, v prípade, že nám bol zarezervovaný laboratórny prvok tejto lokality. Súčasťou konzolového servera je i mechanizmus pre overenie oprávnenosti užívateľových požiadaviek, zabezpečuje prvky pred neoprávneným prístupom a blokuje potencionálne nebezpečné príkazy.

Komunikačný protokol serveru možno rozdeliť na dve fázy:

1.3.1 Inicializácia komunikácie

Klient požadujúci pripojenie na niektorý laboratórny prvok kontaktuje aplikáciu konzolového servera, načúvajúcu na porte 10000. Po úspešnom nadviazaní kontaktu očakáva konzolový server postupnosť štyroch riadkov, ukončených znakom LF, prípadne CRLF. Riadky sú nasledovné:

- Názov zariadenia vo formáte id@lokalita
- PHP session id prihláseného užívateľa, alebo špeciálny reťazec ak sa nejedná o prihláseného užívateľa
- Tutor mode charakterizujúci spôsob prístupu na prvok a uzamykanie konzolí pre ostatných užívateľov
- ID rezervácie v tvare id@lokalita

1.3.2 Beh

Po úspešnej autorizácii server prijíma znaky a posiela ich na konzolu zariadenia, výstup konzoly posiela na výstup užívateľa. Žiadne ďalšie riadiace znaky nie sú definované.

2 Generátory toku

Generátor toku v počítačovej sieti možno charakterizovať ako aplikáciu vytvárajúcu dátové prenosy na sieti. Nemusí sa pritom jednať o dáta zmysluplné, môžu sa prenášať i dáta náhodné. Musia však simulovať dátové prenosy, ktoré sa reálne v sieťach nachádzajú. Teda musia používať niektoré z protokolov vrstiev OSI modelu.

2.1 Požiadavky na generátory

Keďže hlavným účelom systému Virtlab je simulovať reálnu počítačovú sieť, obdobné požiadavky budú mať i dátové toky v sieti systému. Snaha je teda vytvoriť podobné toky aké možno nájsť v sieťach menších rozmerov, až po siete rozmerov Internetu.

Požiadavky na generátory:

- Simulovať čo najvernejšie dátové toky reálnych sietí
- Schopnosť generovať dátové toky rôznych protokolov, rôznych vrstiev OSI modelu
- Možnosť sledovať a vyhodnocovať toky
- Schopnosť ovládateľnosti prostredníctvom rozhraní poskytnutých systémom Virtlab

Pretože sa snažíme vytvoriť reálne toky, je logickou úvahou využívať aplikácie, ktoré takéto toky vytvárajú pri svojej bežnej funkcii. Teda napr. ak sa snažíme vytvoriť na sieti tok, ktorý by bol na reálnej sieti vytvorený prenosom súboru cez protokol FTP, je logické použiť pre jeho vytvorenie FTP server a FTP klienta, a iniciovať prenos súboru medzi nimi. Z požiadaviek sú však zjavné i obmedzenia takýchto generátorov.

Obmedzenia generátorov:

- Spustiteľnosť na prvkoch systému Virtlab, vzhľadom na operačný systém prvkov
- Spustiteľnosť na prvkoch, vzhľadom na technické možnosti prvkov
- Ovládateľnosť generátorov cez konzolu
- Licenčné podmienky umožňujúce využitie v systéme Virtlab

Potrebujeme teda generátory, schopné bežať pod operačným systémom Linux pre virtuálne prvky typu PC a generátory schopné bežať na platforme IOS pre prvky typu router a switch. Ďalej je potrebné aby boli generátory ovládateľné cez fyzickú konzolu zariadení, teda ovládacie rozhranie nemôže byť grafické a všetky vstupy a výstupy musia byť spracovateľné zo štandardných vstupov, výstupov a prípadne filesystému. Prevádzkové podmienky umožňujú použiť iba generátory s licenciou typu freeware a opensource.

Je jasné, že aplikácia, spĺňajúca všetky požiadavky a obmedzenia, neexistuje a naprogramovať ju by bolo veľmi obtiažne. Najvhodnejším variantom je skombinovanie existujúcich aplikácií a vytvorenie jednotného rozhrania pre ovládanie.

Ako bolo povedané skôr, najvýhodnejšou voľbou je voľba „reálnych“ aplikácií, vytvárajúcich toky – trafficy pri svojej bežnej činnosti, pretože budú určite autentickou zložkou skutočných tokov. Architektúra Virtlab však neumožňuje použitie výhradne takýchto aplikácií, ak sa snažíme pokryť čo najväčšie množstvo protokolov. Typický príklad predstavuje dátový tok VoIP telefónie, kedy reálna aplikácia očakáva vstup zo zvukovej karty a výstup taktiež odosiela na zvukovú kartu. Trafficy takýchto aplikácií teda možno generovať iba ako simulovaný.

Generátory trafficu budú teda zastupovať aplikácie s charakterom reálnych aplikácií alebo simulátorov. V nasledujúcich častiach budú popísaní najvhodnejší kanditáti, spĺňajúci podmienky pre integráciu do systému Virtlab.

2.2 Simulátory

Princíp funkčnosti simulátorov je jednoduchý. V podstate ide o odosielanie náhodne generovaných dát protokolmi, ktoré používajú reálne aplikácie. Hlavným problémom simulátorov je, že nie vždy je traffic totožný s trafficom reálnej aplikácie. Nemusia súhlasiť všetky parametre protokolu s čím môže mať problém napríklad stavový firewall, neschopný rozoznať prebiehajúci session.

Ako zástupcovia simulátorov boli pre potreby systému Virtlab vybraní *D-ITG (Distributed Internet Traffic Generator) a SIPp.*

2.2.1 D-ITG – Distributed Internet Traffic Generator

D-ITG predstavuje multi-protokolový a dnes už i multi-platformový traffic generator. Projekt vznikol ako opensource pod záštitou Neapolskej Univerzity. Súčasná verzia 2.7.0-Beta2 podporuje generovanie tokov nasledujúcich protokolov:

- Transportná vrstva : TCP, UDP, DCCP, SCTP
- Aplikačná vrstva : DNS, Telnet, VoIP (kodeky : G.711.1, G.711.2, G.723.1, G.729.2, G.729.3, protokoly : RTP, CRTP), Counter-Strike, Quake 3

Aplikácia sa skladá zo šiestich modulov, pre použitie v systéme Virtlab sú dôležité predovšetkým nasledujúce štyri:

ITGSend – Modul pre odosielanie generovaných dát

ITGRecv – Modul pre prijímanie dát. Niektoré protokoly, potrebujú pre svoju funkciu spoluprácu oboch strán komunikácie. Typickým príkladom je protokol TCP, u ktorého je každý paket obojstranne potvrdzovaný.

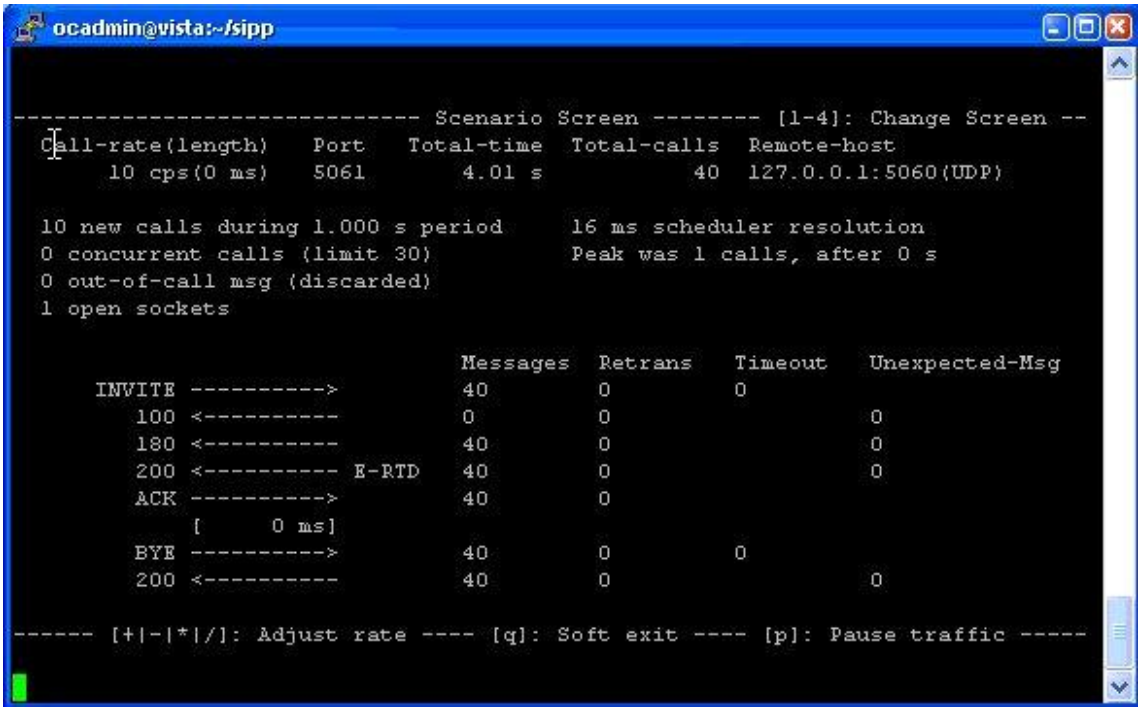
ITGLog – Modul pre vzdialené logovanie. Užívateľ môže nechať preposielať logy z ITGSend a ITGRecv na vzdialený stroj, kde ich spracováva práve tento modul.

ITGDec – Modul pre spracovanie logov a zobrazenie zhrnutia v textovej podobe.

Všetky moduly sa konfigurujú pomocou prepínačov príkazovej riadky. Zoznam všetkých nastavení je dostupný v dokumentácii projektu[1]. Zaujímavý je najmä fakt, že všetky parametre toku sa zadávajú na strane odosielateľa. Modul pre prijímanie je potrebné na strane príjemcu iba spustiť, všetka konfigurácia je prenesená od odosielateľa. Výhodou je taktiež možnosť zobrazenia zhrnutia v textovej podobe, keďže systém Virtlab umožňuje komunikáciu práve cez textovú konzolu.

2.2.2 SIPp

Ako napovedá názov tejto aplikácie, jedná sa o opensource generátor trafficu pre signalizačný protokol SIP. SIPp jednak simuluje signalizáciu sip, ale dokáže odosielať aj RTP dáta zo súboru PCAP, ktorý je možné vytvoriť z reálneho hovoru napríklad cez aplikáciu Wireshark. Tie môžu byť vo formáte audio resp. audio + video. Prijaté dáta sú potom odosielané späť odosielateľovi. Jedná sa teda o plnohodnotnú simuláciu VoIP hovoru.



Obrázok 2

[1] http://www.grid.unina.it/software/ITG

Konfiguráciu predstavuje nadefinovanie scenára pomocou XML súboru. Prostredníctvom konzoly možno potom aplikáciu spustiť s požadovaným scenárom. Naskytá sa i možnosť vytvorenia skriptu, ktorý bude prepisovať parametre príkazovej riadky na tagy v XML súbore, ktorý podstrčí hlavnej aplikácii. Zhrnutie hovoru je vypisované v textovej podobe na štandardný výstup viď obrázok 2. Dokumentáciu možno nájsť na stránkach projektu[2].

2.3 Reálne aplikácie

Najväčším problémom, ktorý majú reálne aplikácie je fakt, že pre svoju činnosť potrebujú prenášať reálne dáta. Nie je však možné uchovávať na prvkoch systému Virtlab veľké množstvo dát. Hlavná príčina spočíva najmä v skutočnosti, že prvky bežia na virtuálnom stroji fyzicky na jednom počítači. Teda ak by každý prvok mal uchovávať veľké množstvo dát, bolo by potrebné úložisko s veľkosťou väčšou ako súčet veľkostí všetkých veľkých súborov. Generovanie súborov taktiež neprichádza do úvahy, pretože spustenie generátorov na viacerých prvkov by viedlo k rozdeleniu diskového a procesorového času na fyzickom stroji a dáta by nemuseli byť generované dostatočne včas. Problém vzniká i pri aplikáciách, ktoré vyžadujú kópiu toho istého súboru na viacerých prvkoch. Problém čiastočne rieši uchovávanie veľkých súborov v spoločnom úložisku pre všetky virtuálne stroje, teda na fyzickom úložisku by sa nachádzal iba raz. Pri návrhu úloh však treba i tak počítať s faktom, že všetky virtuálne prvky používajú to isté úložisko pre načítanie i ukladanie súborov.

Ako zástupcovia generátorov toku boli vybrané nasledujúce aplikácie:

2.3.1 Web server + klient

Pre serverovú časť postačuje ľubovoľný web server s podporou protokolov http a https, napr. apache[3]. Konfiguráciu serveru predstavuje spustenie s nasmerovaním k súboru v zdieľanom úložisku. Všetky prichádzajúce dáta od užívateľa budú zahodené.

Klient je obvykle zahrnutý ako súčasť operačného systému. V systéme Linux ho predstavuje aplikácia wget. Prijaté súbory sa taktiež zahadzujú.

2.3.2 FTP server + klient

Obdobne ako pri web serveri, postačuje ľubovoľný server s podporou protokolu ftp, napr. vsftpd[4]. Konfigurácia spočíva vo vytvorení anonymného účtu umožňujúceho stiahnutie súboru zo zdieľaného úložiska. Prichádzajúce dáta sa zahadzujú.

Ako klienta je taktiež možné použiť aplikáciu wget. Pre možnosť odosielania sú dostupné aplikácie konfigurovateľné z konzoly, napr. ftpput[5]. Všetky prichádzajúce dáta sú taktiež zahadzované.

[2] http://sipp.sourceforge.net/
[3] http://www.apache.org/
[4] http://vsftpd.beasts.org/

2.3.3 SSH server + klient

Tento typ trafficu je špecifický predovšetkým pre systémy založené na OS Linux. Server býva integrovaný v systéme ako démon sshd. Konfiguráciu predstavuje vytvorenie účtu v systéme s právom čítania súboru v zdieľanom úložisku a výmenu certifikátov medzi klientmi a serverom, pre zamedzenie požiadaviek na heslo.

Klientskú časť predstavuje integrovaná aplikácia scp. Všetky prichádzajúce dáta sa zahadzujú obdobne ako u predchádzajúcich typov tokov.

2.3.4 Ping

Štandardný nástroj pre zistenie dostupnosti siete a prvkov prostredníctvom ICMP protokolu. Nevyžaduje server. Klient je obsiahnutý v každom operačnom systéme. Predbežná konfigurácia nie je potrebná.

2.3.5 Bit Torrent

Torrent na rozdiel od predchádzajúcich aplikácií nemá charakter serveru a klienta. Keďže sa súbory sťahujú od viacerých klientov súčasne a zároveň sa odosielajú ďalším, má bit torrent traffic charakter skôr P2P architektúry. V sieti existujú riadiace body, trackery, slúžiace na výmenu informácií medzi klientmi, nie však sťahovaných dát. Aby sa klient dostal k súborom, ktoré požaduje od ostatných, potrebuje metasúbor s koncovkou .torrent. Ten vytvoril prvý klient poskytujúci súbor pre stiahnutie, z tohto hľadiska sa dá chápať ako server. Prenos súboru však pokračuje i po odpojení servera.

Najvhodnejšieho kandidáta predstavuje na Linuxe aplikácia BitTornado[6]. Obsahuje nástroje pre vytvorenie klienta, trackeru i metasúboru. Konfiguráciu predstavuje vytvorenie verejného trackeru, týchto trackerov môže byť obecne neobmedzený počet. Pre vytvorenie metasúboru je potrebné vedieť adresy trackerov. Keďže v systéme Virtlab nie je spôsob, ako rozdistribuovať metasúbor zo serveru na ostatných sťahujúcich klientov, je potrebné súbory vygenerovať na každom klientovi zvlášť pri jeho spúšťaní. Keďže majú všetci klienti prístup k prenášanému súboru v zdieľanom úložisku, generujú pre tento súbor zhodný metasúbor. V prípade, žeby nebolo možné vygenerovať identický metasúbor, je možné metasúbor editovať textovo, keďže adresy trackerov sú v ňom uložené v textovej podobe. Po vygenerovaní metasúboru s aktuálnymi adresami trackerov už postačuje iba predhodiť tento klientom.

Najväčším problémom zostáva fakt, že súbor je používaný počas celého procesu sťahovania, pretože hociktorý klient môže požiadať o jeho časť. Sťahovaný súbor preto nemôže byť zahodený ako tomu bolo u predchádzajúcich typov trafficov. To znamená, že každý stroj si uchováva vlastnú kópiu sťahovaného súboru. Je preto potrebné myslieť na fyzické obmedzeniach stro-

[5] http://www.unixwiz.net/tools/
[6] http://www.bittornado.com/

ja, na ktorom bežia prvky, predovšetkým na kapacitu a rýchlosť pevného disku zariadenia. Taktiež nemožno zabudnúť na údržbu po dokončení toku, teda na vymazanie takýchto kópií.

Tento typ trafficu bol vybraný predovšetkým pre jeho vlastnosti a rozšírenosť. V roku 2009 bolo vypočítané, že 43% celkového dátového toku v internete je práve typu bit torrent. Špecifikom je i protokol bit torrentu, vytvárajúci tisíce spojení na rôznych klientov. Práve toto je dôvod, prečo je u mnohých ISP neobľúbený, zahlcuje databázy sieťových prvkov a spôsobuje výpadky. Pri svojej rozšírenosti s ním však treba počítať ako so štandardným zástupcom tokov na sieti.

3 Analýza a návrh

Táto kapitola pojednáva o analýze a návrhu systému, predstavujúceho rozhranie pre prácu s generátormi, popísanými v predchádzajúcej kapitole.

3.1 Analýza požiadaviek

Pri analyzovaní požiadaviek hľadáme odpovede na otázky.

Dôvod pre tvorbu nového systému (Prečo?)

Potrebujeme systém, ktorý umožní ovládať generátory trafficu v systéme Virtlab. Keďže pre ne nie je dostupné žiadne užívateľské rozhranie, potrebujeme navrhnúť také, ktoré bude jednak zapadať do konceptu systému Virtlab a bude podobné natívnemu ovládaniu generátorov. Ďalej potrebujeme schopnosť systému nastavenia generátorov ukladať a použiť opakovane. Vhodná je aj možnosť plánovania spúšťania tokov. Systém nesmie byť viazaný len na použitie preddefinovaných generátorov, ale musí byť schopný prispôsobeniu iným, prípadne zmene verzie a syntaxe generátorov. Systém taktiež nesmie byť viazaný na operačné systémy použité na prvkoch topológie.

Užívatelia systému (Kto?)

Systém budú obsluhovať jednoduchí užívatelia – študenti, ktorý budú môcť využívať funkcie systému a vytvárať toky na sieti. Špeciálny užívateľ „správca úlohy“ bude môcť, okrem svojich vlastných tokov, definovať i toky preddefinované, viditeľné a použiteľné ostatnými užívateľmi.

Vstupy

- Vstupom do systému od užívateľa sú parametre zadané pre jednotlivé toky. Predovšetkým parametre konkrétneho toku, čas a dátum spustenia, prípadne pokyny pre spustenie, zastavenie, vymazanie alebo vytvorenie nového toku.
- Vstupom do systému zo strany správcu úloh sú všetky vstupy bežného užívateľa. Navyše môže dávať pokyny pre zaradenie tokov do preddefinovaných, prípadne vyradenie.
- Vstupom do systému z hľadiska konfigurácie budú konfigurácie parametrov jednotlivých generátorov, ako cesty k spúšťačom a zoznamy konfigurovateľných atribútov. Taktiež to budú konfigurácie pre rôzne typy operačných systémov, konkrétne postup spúšťania generátorov na konkrétnom OS.

Výstupy

- Zoznam definovaných tokov a ich stav
- Zoznam preddefinovaných tokov a ich stav
- Zoznam konfigurovateľných parametrov jednotlivých typov tokov

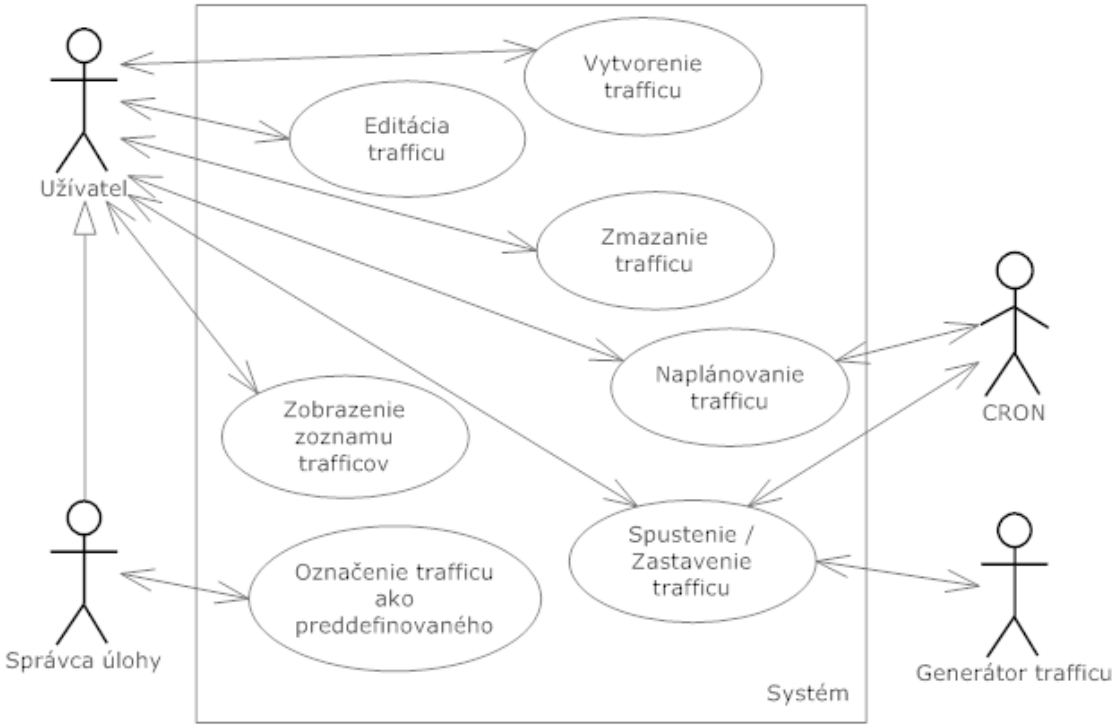
- Zoznam všetkých typov tokov, ktoré môžu byť generované na zvolenom zariadení
- Zoznam tokov zlúčených pre vytvorenie kombinovaných tokov
- Spúšťacie reťazce odosielané na konzolu zariadenia, z ktorého sa budú generovať toky
- Spúšťacie reťazce pre plánované spúšťanie tokov, ktoré sa odosielajú na CRON
- Výstupné štatistiky generovaných tokov

Okolie systému

Okolie systému predstavujú užívatelia systému, generátory trafficu na sieťových prvkoch topológie a server pre spúšťanie naplánovaných tokov CRON.

Nefunkčné požiadavky

Systém by mal byť prehľadný a jednoduchý na ovládanie. Mal by zapadať do kontextu rozhrania systému Virtlab. Taktiež by mal byť viacjazyčný. Pre vývoj je použitý jazyk PHP spolu so SRBD MySQL.

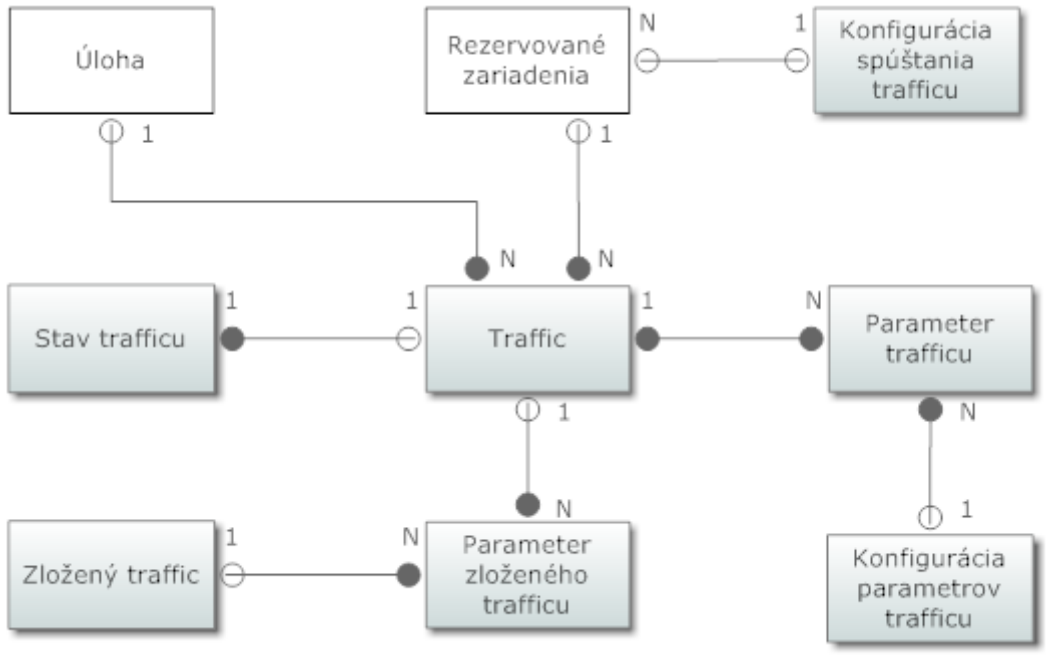


Obrázok 3

Celkový prehľad funkčnosti z analýzy požiadaviek je zjavný z diagramu prípadov užitia na obrázku 3.

3.2 Dátová analýza

Z požiadaviek vyplýva, že bude potrebné uchovávať informácie o trafficoch vytvorených užívateľmi, ďalej parametre trafficov. Potrebné je taktiež uchovávať aktuálny stav trafficu. Užívateľ má možnosť vytvárať zložené trafficy z elementárnych, takže je potrebné uchovávať skladbu takýchto multitrafficov. V neposlednom rade treba ukladať konfigurácie pre spúšťanie trafficov na rôznych operačných systémoch a ukladať i popisy generátorov a ich možné nastavenia. Na obrázku 4 vidno interakcie medzi jednotlivými uchovávanými entitami na ER diagrame. Na obrázku taktiež vidno napojenia na uchovávané entity systému Virtlab, označené odlišnou farbou.



Obrázok 4

Systém generátoru trafficu teda bude obsahovať sedem tabuliek. V nich budú uchovávané nasledujúce atribúty.

Traffic:

- traffic_id – identifikátor trafficu, kľúč
- creator – login užívateľa, ktorý vytvoril tento traffic
- task_id – úloha, pre ktorú je tento traffic vytvorená, cudzí kľúč z tabuľky Úloha
- name – pomenovanie trafficu
- config – meno konfigurácie, ktorá bola použitá pre definovanie parametrov trafficu, cudzí kľúč z tabuľky Konfigurácia parametrov trafficu

- vertex – názov zariadenia, pre ktoré je tento traffic určený, cudzí kľúč z tabuľky Rezervované zariadenia
- private – identifikácia, či sa jedná o súkromný traffic, alebo preddefinovaný správcom úlohy

Stav trafficu:

- id – identifikátor stavu trafficu, kľúč
- creator – užívateľ, ktorý zmenil stav trafficu
- traffic_id – id trafficu, ktorého stav je zachytený, cudzí kľúč z tabuľky traffic
- res_id – id rezervácie cudzí kľúč z tabuľky Rezervované zariadenia
- from – čas pre plánované spustenie trafficu
- to – čas pre plánované zastavenie trafficu
- running – indikuje, či je traffic práve beží
- vertex – názov zariadenia, pre ktoré je traffic určený, cudzí kľúč z tabuľky Rezervované zariadenia
- additional – doplňujúce informácie o bežiacom trafficu, potrebné napríklad pre jeho zastavenie

Parameter trafficu:

- traffic_id - id trafficu, ktorého stav je zachytený, cudzí kľúč z tabuľky traffic
- advanced – indikuje, či sa jedná o rozšírený, alebo základný parameter trafficu
- index – poradové číslo parametra, cudzí kľúč z tabuľky Konfigurácia parametrov trafficu
- value – hodnota parametra

prvé tri atribúty predstavujú kľúč

Zložený traffic:

- id – identifikátor zloženého trafficu, kľúč
- creator – užívateľ, ktorý vytvoril kombinovaný traffic
- task_id – úloha, pre ktorú je tento traffic vytvorená, cudzí kľúč z tabuľky Úloha
- name – pomenovanie zloženého trafficu
- private – identifikácia, či sa jedná o súkromný traffic, alebo preddefinovaný správcom úlohy

Parameter zloženého trafficu:

- multischedule_id – id zloženého trafficu, ktorému parameter prináleží, cudzí kľúč z tabuľky Zložený traffic

- traffic_id – id trafficu, ktorý je časťou zloženého trafficu, cudzí kľúč z tabuľky traffic
- delay – oneskorenie spustenia trafficu
- delay_end – oneskorenie zastavenia trafficu

Konfigurácia parametrov trafficu:

- generator_os – OS na ktorom je možné generovať druh tohto typu trafficu
- generator_id – indentifikátor konfigurácie parametrov trafficu, kľúč
- generator_title – názov typu trafficu
- generator_pathsender – cesta k spustiteľnému súboru generátoru
- generator_parameters – vymenovanie parametrov generátora, ktoré možno konfigurovať
- generator_advancedparameters – vymenovanie pokročilých parametrov generátora, ktoré možno konfigurovať
- generátor_always – parametre, ktoré sa dopisujú vždy

Konfigurácia spúšťania trafficu:

- os – OS pre ktorý platí spúšťanie týmto touto konfiguráciou, kľúč
- trafficstart – popis procesu spustenia trafficu v programovacom jazyku
- trafficstop – popis procesu zastavenia trafficu v programovacom jazyku
- analyzator – popis pre zobrazenie výsledkov o trafficu užívateľovi

Keďže posledné dve entity nie je potreba často obmieňať a nachádzajú sa v nich i kódy v programovacom jazyku a jazykoch konzolí operačných systémov, bude výhodnejšie realizovať ich ako súbory, než ako tabuľky databáze. Parametre je potom možné realizovať ako premenné v jazyku, v ktorom je napísaný systém, v tomto prípade PHP.

3.3 Návrh funkcií

Po zvážení požiadaviek boli navrhnuté nasledovné funkcie systému. Jednotlivé typy užívateľov sú značené písmenami U – bežný užívateľ, S – správca úlohy, C – CRON. Správca úlohy už pritom má práva bežného užívateľa.

Id	Prístup	Kľúčové slovo	Funkcia	Priorita
1	U	Zobraz trafficy užívateľa	ListTraffic	MUST HAVE
2	U	Zobraz zložené trafficy užívateľa	ListMultitraffic	MUST HAVE
3	U	Zobraz preddefinované trafficy	ListPredefTraffic	MUST HAVE
4	U	Zobraz preddefinované zložené trafficy	ListPredefMultitraffic	MUST HAVE

5	U	Zobraz trafficy k ostatným úlohám	ListOtherTraffic	COULD HAVE
6	U	Kopíruj traffic	CopyTraffic	MUST HAVE
7	U	Edituj traffic	EditTraffic	MUST HAVE
8	U	Zmaž traffic	DeleteTraffic	MUST HAVE
9	S	Kopíruj preddefinovaný traffic	CopyPublicTraffic	MUST HAVE
10	S	Edituj preddefinovaný traffic	EditPublicTraffic	MUST HAVE
11	S	Zmaž preddefinovaný traffic	DeletePublicTraffic	MUST HAVE
12	U	Kopíruj traffic z preddefinovaných	CopyToPrivate	MUST HAVE
13	S	Kopíruj traffic do preddefinovaných	CopyToPublic	MUST HAVE
14	U	Zmaž traffic k inej úlohe	DeleteOtherTraffic	COULD HAVE
15	U	Vytvor nový traffic	CreateTraffic	MUST HAVE
16	U	Vytvor zložený traffic	CreateMultitraffic	MUST HAVE
17	U	Pridaj traffic do zloženého trafficu	AddTraffic	MUST HAVE
18	U	Odober traffic zo zloženého trafficu	RemoveTraffic	MUST HAVE
19	U	Edituj zložený traffic	EditMultitraffic	MUST HAVE
20	U	Vymaž zložený traffic	DeleteMultitraffic	MUST HAVE
21	S	Vytvor preddefinovaný zložený traffic	CreatePublicMultitraffic	MUST HAVE
22	S	Pridaj traffic do preddefinovaného zloženého trafficu	AddPublicTraffic	MUST HAVE
23	S	Odober traffic z preddefinovaného zloženého trafficu	RemovePublicTraffic	MUST HAVE
24	S	Edituj zložený preddefinovaný traffic	EditPublicMultitraffic	MUST HAVE
25	S	Vymaž zložený preddefinovaný traffic	DeletePublicMultitraffic	MUST HAVE
26	U	Kopíruj zložený traffic	CopyMultitraffic	MUST HAVE
27	U	Kopíruj zložený traffic z preddefinovaných	CopyMultiToPrivate	MUST HAVE
28	S	Kopíruj zložený preddefinovaný traffic	CopyPublicMultitraffic	MUST HAVE
29	S	Kopíruj zložený traffic do preddefinovaných	CopuMultiToPublic	MUST HAVE
30	U	Naplánuj traffic	ScheduleTraffic	MUST HAVE
31	U	Spusti traffic	StartTraffic	MUST HAVE
32	U,C	Zastav traffic	StopTraffic	MUST HAVE
33	C	Spusti naplánovaný traffic	StratScheduledTraffic	MUST HAVE
34	U	Naplánuj zložený traffic	ScheduleMultitraffic	MUST HAVE

35	U	Zobraz štatistiky trafficov	ShowStats	MUST HAVE
Celkom 32				

3.4 Popis neelementárnych funkcií

Väčšina funkcií predstavuje triviálne operácie vytvorenia, editácie a mazania záznamov. V návrhu sa však nachádzajú aj netriviálne. Niektoré, demonštračné, budú v tejto kapitole bližšie popísané v podobe minišpecifikácií (pseudokód), dátové toky budú vyobrazené v podobe DFD diagramov. Zvyšné netriviálne funkcie predstavujú menšiu obmenu, prípadne podmnožinu popísaných.

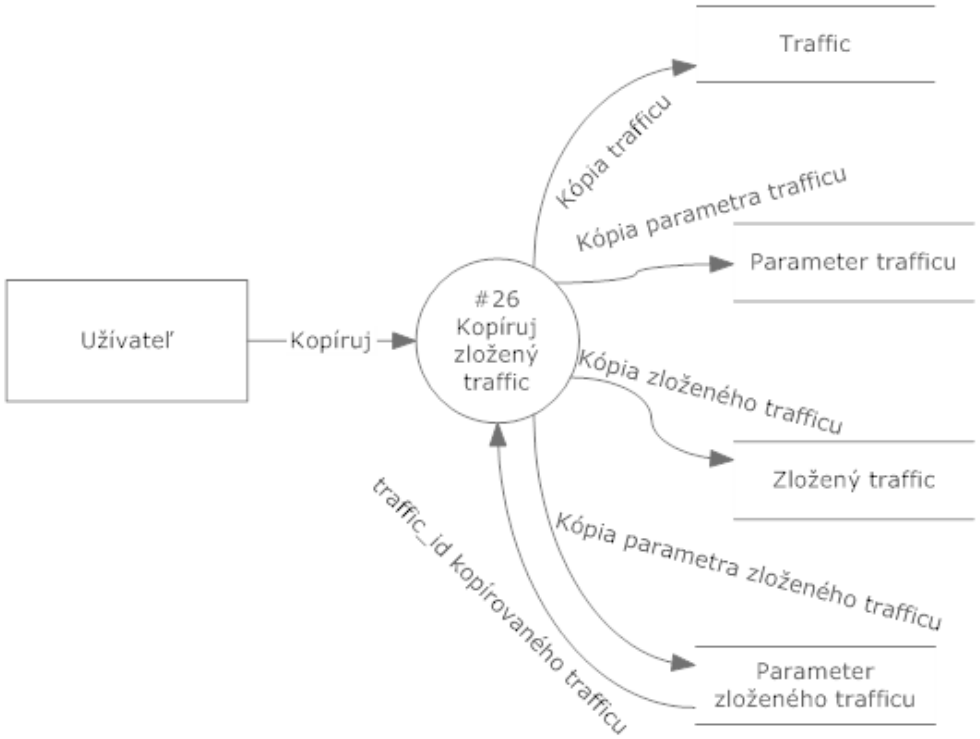
3.4.1 CopyMultitraffic

funkcia #26 Kopíruj zložený traffic

Minišpecifikácia:

1. Zobraz formulár pre výber zloženého trafficu pre skopírovanie
2. Vyber zložený traffic, ktorého id zodpovedá zvolenému a zapamätaj ho do P
3. Zobraz formulár pre pomenovanie nového zloženého traffic a výsledok ulož do L
4. Vytvor nový zložený traffic, ktorého atribút creator = aktuálne prihlásený užívateľ, task_id = P.task_id, private=0, name=L a zapamätaj si jeho id do M
5. Vyber Parametre zloženého trafficu, ktorých multischedule_id = P.id a ulož ich do poľa P
6. Pre každý prvok v P vykonaj:
    1. Vyber Traffic, ktorého traffic_id = P.traffic_id a ulož ho do L
    2. Vytvor nový Traffic s parametrami z L a zapamätaj si jeho traffic_id do N
    3. Vytvor Parameter zloženého trafficu s atribútmi traffic_id = N, multischedule_id = M
    4. Vyber Parametre trafficu, ktorých traffic_id = L a ulož ich do poľa O
    5. Pre každý prvok v O vykonaj:
        1. Vytvor nový Parameter trafficu s atribútami traffic_id = N, advanced = O.advanced, index = O.index, value = O.value
7. Zobraz výsledok

DFD diagram na obrázku 5 zobrazuje prevládajúce dátové toky. Funkcia prijíma ako vstup požiadavku o skopírovanie zloženého trafficu od užívateľa. Potrebuje zistiť z akých všetkých trafficov sa kopírovaný zložený traffic skladá a následne kopíruje záznamy.



Obrázok 5

3.4.2 ScheduleTraffic

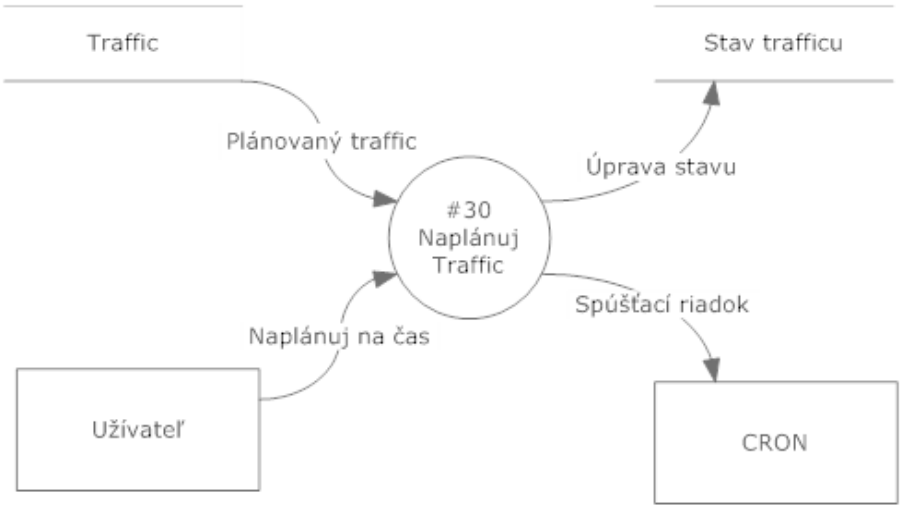
funkcia #30 Naplánuj traffic

Minišpecifikácia:

1. Zobraz formulár pre plánovanie trafficu a ulož id vybraného trafficu do P.id, čas začiatku do P.from, čas konca do P.to
2. Vyber Traffic, ktorého traffic_id = P.id a ulož ho do L
3. Vyber Stav trafficu, ktorého vertex = L.vertex a (abs(from-P.from) < 15, alebo abs(to-P.from) < 15, alebo abs(from-P.to) < 15, alebo abs(to-P.to) < 15)
4. Ak existuje skoč na 7.
5. Vytvor stav trafficu s atribútmi from = P.from, to = P.to, running = 0, creator = L.creator, vertex – L.vertex, res_id = id aktuálnej rezervácie a ulož jeho id do M
6. Vytvor riadok pre CRON s atribútmi creator = L.creator, id = M a odošli ho CRONu
7. Zobraz výsledok

Bod 3. minišpecifikácie predstavuje kontrolu, či na prvku na ktorom sa snažíme naplánovať traffic neprebieha práve spúšťanie alebo zastavovanie iného trafficu. Číslo 15 reprezentuje ochranný časový interval medzi rezerváciami, kedy môže pristupovať na prvok iný proces.

DFD diagram na obrázku 6 znázorňuje prijatie požiadavku od užívateľa, kde proces zisťuje detaily o trafficu. Následne je aktualizovaný stav trafficu a odoslaná požiadavka na CRON server, pre oneskorené spustenie trafficu.



Obrázok 6

3.4.3 StartScheduledTraffic

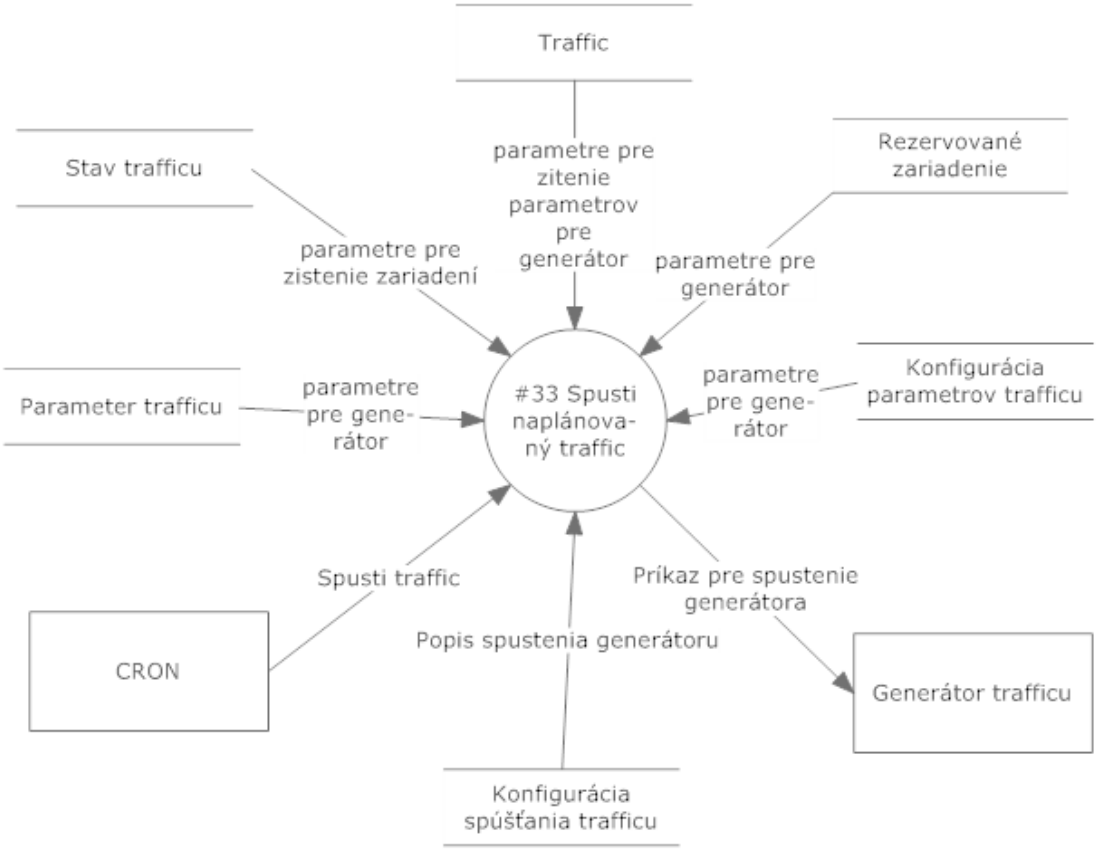
funkcia #33 Spusti naplánovaný traffic

Minišpecifikácia:

1. Príjmi požiadavku od CRONu a ulož z nej atribút id do P
2. Vyber Stav trafficu, ktorého id = P a ulož ho do R
3. Vyber Traffic, ktorého traffic_id = R.traffic_id a ulož ho do S
4. Vyber Parametre trafficu, ktorých traffic_id = R.traffic_id a ulož ich do T
5. Vyber Konfiguráciu parametrov trafficu, ktorej generator_id = S.config a ulož ju do U
6. Skopíruj príkaz pre spustenie z U.generator_pathsender do V
7. Pre každý záznam z U.generator_parameters vykonaj:
    1. Vyber záznam z U.generator_parameters a pridaj ho reťazca V
    2. Vyber záznam z T.advanced = 0 a T.index = aktuálny index v U.generator_parameters a pridaj ho do reťazca V
8. Pre každý záznam z U.generator_advancedparameters vykonaj:
    1. Ak neexistuje záznam s T.advanced = 0 a T.index = aktuálny index v U.generator_advanced parameters skoč na 9.
    2. Vyber záznam z U.generator_advancedparameters a pridaj ho reťazca V
    3. Vyber záznam z T.advanced = 1 a T.index = aktuálny index v U.generator_advancedparameters a pridaj ho do reťazca V

9. Vyber záznam Rezervované zariadenie, ktorého atribúty vertex = S.vertex a resid = R.resid a ulož do W
10. Vyber Konfiguráciu spúšťania trafficu, ktorej atribút os = OS zariadenia a ulož starttraffic do O
11. Pripoj sa na konzolový server a odošli údaje W.device, R.resid, R.creator
12. Spusti funkciu zadefinovanú v O a predaj jej string z V
13. Aktualizuj Stav trafficu, ktorého id = P, nastav running = 1

DFD na obrázku 7 zobrazuje zhromaždenie všetkých parametrov potrebných pre spustenie generátoru trafficu. Spúšťanie je zahájené bez interakcie užívateľa, podnet prichádza od CRON servera.



Obrázok 7

4 Implementácia

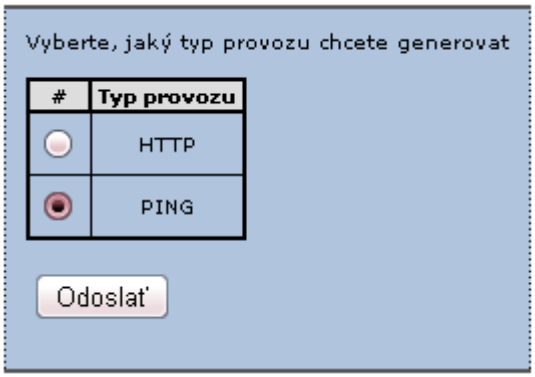
V tejto kapitole sa bližšie pozrieme na implementáciu systému generátoru trafficu pre systém Virtlab. Po prečítaní by čitateľ mal pohopiť používanie generátor trafficu. Taktiež by mal získať prehľad o tom, ako celý systém funguje.

4.1 Užívateľské rozhranie

Ako základ pre užívateľské rozhranie, bolo vybrané rozhranie zobrazujúce aktívne rezervácie užívateľa. Keďže práve tu má užívateľ na očiach všetky veci potrebné pre prácu s topológiou ktorú si zarezervoval, rozšírenie toho rozhrania i na prácu s generátorom tokov je logickou voľbou. Pri prvom spustení si užívateľ všimne, že na tejto obrazovke pribudli pre prvky dve nové funkcie. Pôvodné, prístup na konzolu zariadenia, reštartovanie zariadenia a telnetový prístup, boli rozšírené o voľby Traffic generator a Traffic Stats, viď. obrázok 8.

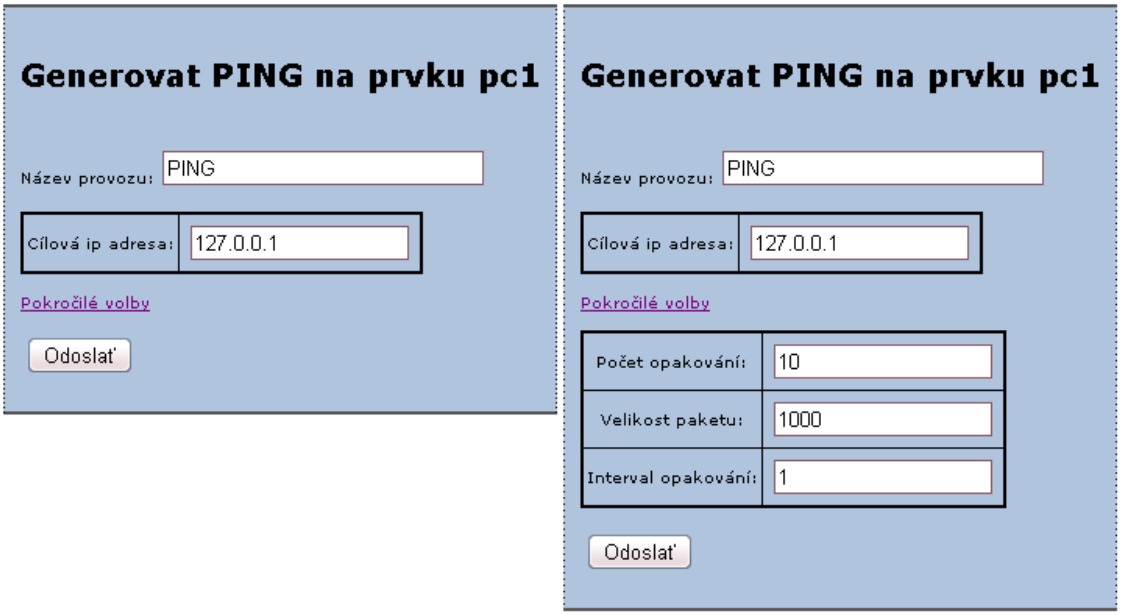
Obrázok 8

Povedzme teda, že chcem generovať nejaký tok na prvku s označením pc1. Klikneme na tlačidlo Traffic generator v príslušnom riadku. Systém následne skontroluje XML, ktorým sa rezervovala topológia a identifikuje z neho operačný systém zariadenia pc1. Ďalšia obrazovka predstavuje prvý krok v nastavení generovaného toku. Systém zistí, aké typy trafficu sa dajú na danom zariadení generovať a prezentuje ich užívateľovi, viď obrázok 9.



Obrázok 9

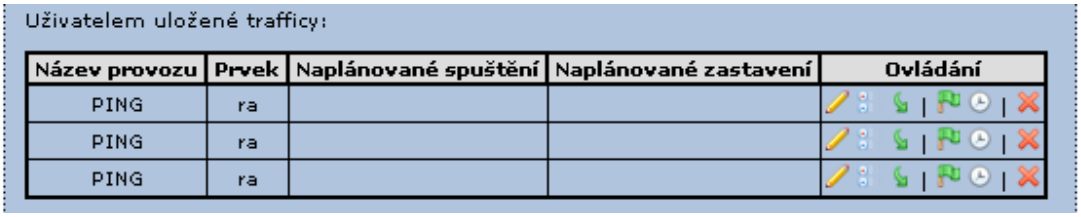
Užívateľ zvolí požadovaný typ trafficu a následne môže konfigurovať parametre, definované v konfigurácii, ktorej sa venuje nasledujúca kapitola.



Obrázok 10

Väčšina trafficov má štandardné a rozšírené nastavenia, ako je to typické u reálnych aplikácií. Niektoré nastavenia je potreba zmeniť, niektoré zriedkavo. Obrázok 10 znázorňuje situáciu zachytenú na trafficu typu PING, kedy je nastavovaná v drvivej väčšine iba cieľová ip adresa stroja, ktorého dostupnosť chceme overiť. Niekedy sa však stane i prípad potreby ďalších, rozšírených nastavení. Táto obrazovka predstavuje druhý a posledný krok nastavenia generovaného toku.

Vytvorené toky sa ukladajú do databázy a sú vždy viazané na konkrétnu úlohu, teda ak užívateľ zarezervuje tú istú úlohu, môže používať toky, ktoré vytvoril. Všetky svoje toky vidí v prehľade na obrazovke aktívnej rezervácie, viď obrázok 11.



Obrázok 11

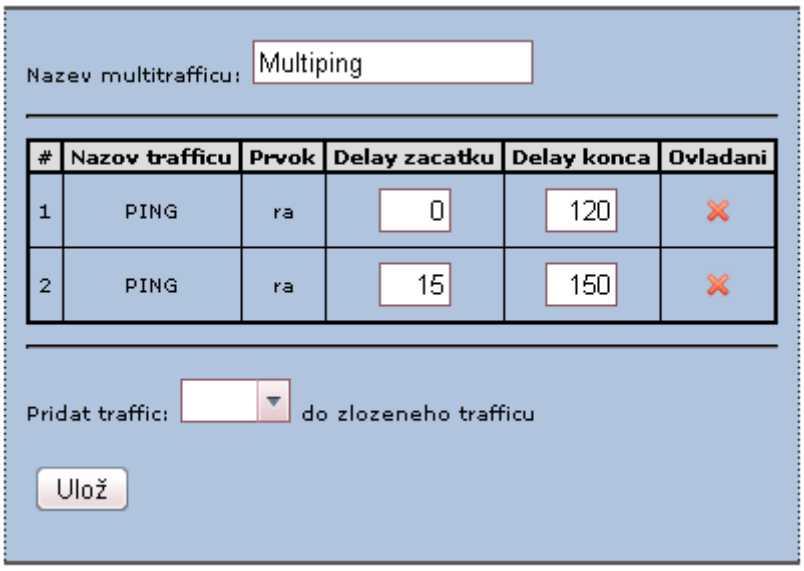
Okrem výpisu svojich vytvorených trafficov vidí užívateľ i trafficy, ktoré boli preddefinované správcom úlohy. V pravej časti výpisu užívateľských i preddefinovaných tokov sa nachádzajú ovládacie prvky pre nasledujúce operácie s nimi:

- - spustenie
- - zastavenie
- - naplánovanie spustenia
- - skopírovanie trafficu z preddefinovaných do užívateľských
- - skopírovanie trafficu z užívateľských do preddefinovaných
- - skopírovanie trafficu
- - editácia trafficu
- - zmazanie trafficu
- - zobrazí doplňujúcu informáciu o trafficu

Užívateľ má možnosť naplánovať úlohu pre neskoršie spustenie. Pre spustenie úlohy potom nemusí byť prihlásený do systému. Užívateľ definuje čas spustenia a zastavenia daného generátoru s presnosťou na sekundu, viď obrázok 11. Tieto časy majú svoje obmedzenia. Prvé obmedzenie obmedzuje naplánovanie času spustenia príliž blízko aktuálnemu času, pretože prvky zodpovedné za plánovanie potrebujú pre túto operáciu určitý čas. Druhé obmedzenie predstavuje odstup operácií spustenia a zastavenia, aby nedošlo k situácii, že na prvku bude prebiehať viacero pokusov o spustenia, alebo zastavenie súčasne. O porušení niektorého z obmedzení je užívateľ informovaný.

Obrázok 12

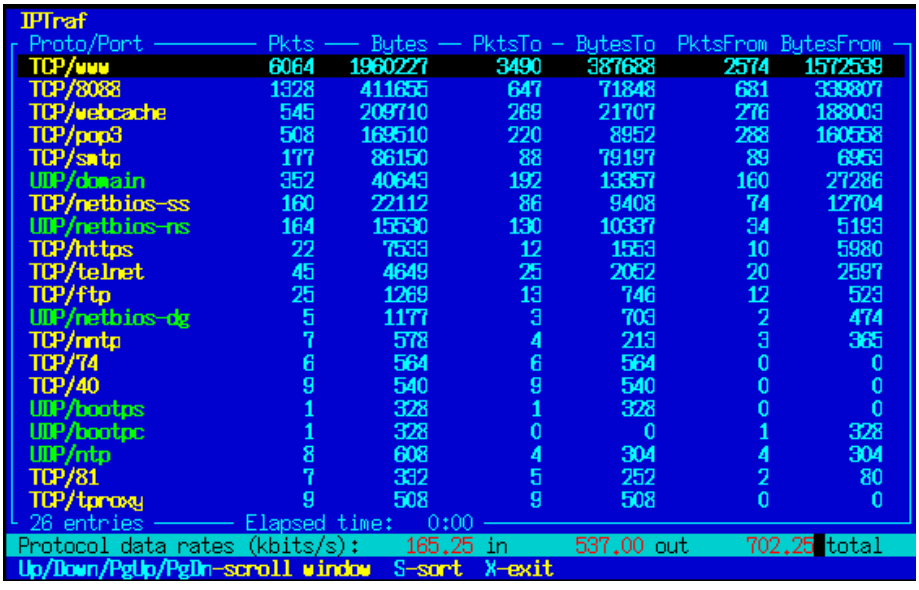
Spájaním trafficov do väčších celkov možno vytvoriť zložený traffic. Výhodou takýchto celkov je možnosť relatívneho plánovania. Teda časy čiastkových trafficov sú nastavené ako oneskorenie spustenie alebo zastavenia trafficu od času spustenia takéhoto zloženého trafficu.



Obrázok 13

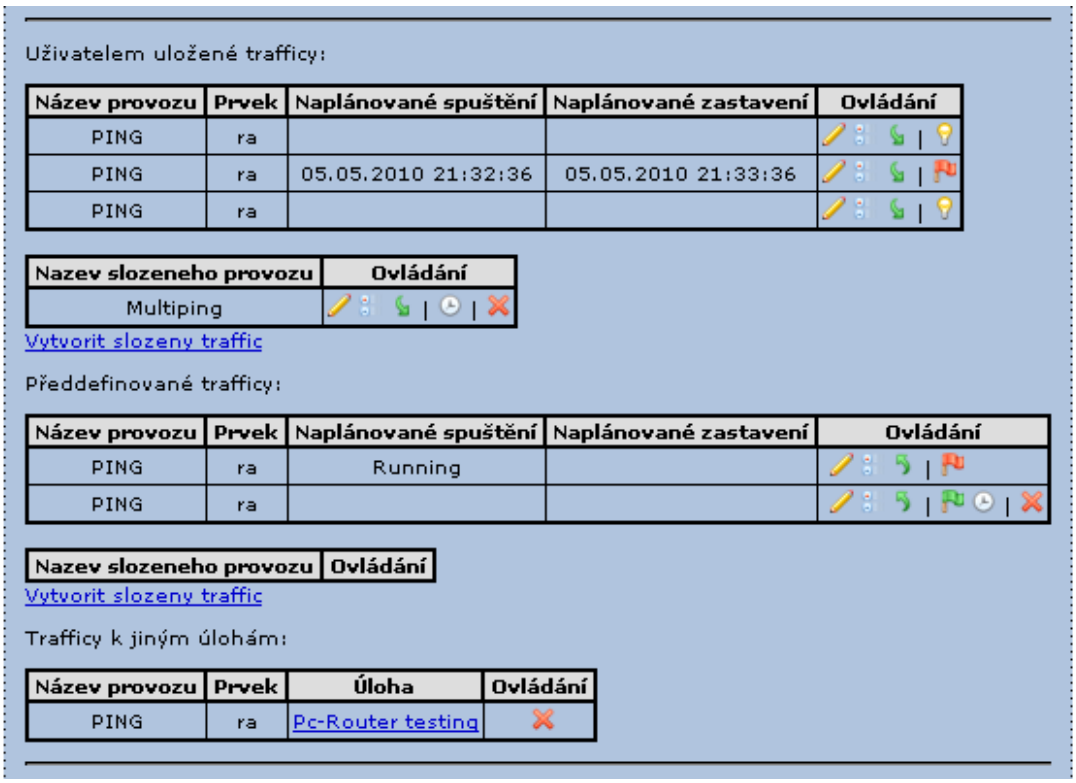
Na obrázku 13 vidno spojenie dvoch trafficov do jedného zloženého trafficu. Pomocou ovládania možno pridávať a odoberať ďalšie elementárne toky. Spúšťanie a zastavovanie je plánované ako oneskorenie, teda traffic s oneskorením 0 bude spustený v čase naplánovaného spustenia.

Tlačidlo Traffic stats vyvolá konzolu zariadenia, do ktorej budú zobrazené výsledky zachytené analyzátormi trafficu. Príklad univerzálneho analyzátoru predstavuje napr. aplikácia IPTraf.



Obrázok 14

Na obrazovke aktívnej rezerváciu budú zobrazené všetky vytvorené trafficy patriace k úlohe i iným úlohám. Taktiež budú zobrazené i preddefinované trafficy a zložené trafficy. Systém zobrazuje ich aktuálny stav a ponúka operácie, ktoré sú v danom stave a s danými právami možné. Celkový prehľad ukazuje obrázok 15.



Obrázok 15

4.2 Implementované moduly

V nasledujúcej časti sú podrobnejšie rozobrané implementované moduly.

4.2.1 reser-active.php

Jedná sa o doplnenie pôvodného modulu systému Virtlab, slúžiaceho predovšetkým na prácu s aktuálne zarezervovanými prvkami topológie. Modul bol doplnený o ovládacie prvky pre vytváranie tokov a zobrazovanie nazbieraných štatistík o tokoch. Modul ďalej zobrazuje trafficy definované užívateľom a trafficy preddefinované správcom úlohy pre aktuálne zarezervovanú úlohu. Pre každý zo zobrazených tokov ponúka ovládacie prvky, ktoré odpovedajú stavu traffi-cu a právam užívateľa.

4.2.2 traffic_generator.php

Modul reprezentuje rozhranie pre vytváranie, editáciu, kopírovanie a mazanie uložených traffi-cov. Modul má tri stavy, fungujúce ako kroky wizarda. V prvom kroku systém vyberá z konfi-

guračných súborov trafficy dostupné pre prvok, na ktorom užívateľ chce traffic generovať a ponúka tieto na výber. Druhý krok predstavuje zadefinovanie parametrov trafficu a jeho pomenovanie. Tretí krok vymazáva a vytvára nové záznamy tabuľky podľa špecifikácie a informuje užívateľa o výsledku.

Niektoré stavy modulu nemusia byť využité v závislosti na operácii, ktorá je požadovaná. Taktiež niektoré operácie v stavoch môžu byť vynechané. Napríklad ak užívateľ zvolí mazanie trafficu, modul vykoná iba stav tri a vymaže odpovedajúce riadky. Ďalší príklad predstavuje požiadavka na kopírovanie, kedy modul začína v druhom stave. Zobrazuje však iba formulár pre pomenovanie nového trafficu, neumožňuje už definovanie parametrov. Následne prechádza do kroku tri a vytvára nové záznamy.

4.2.3 traffic_starter.php

Modul nesúci tento názov predstavuje hlavné jadro funkčnosti systému. Jeho základ predstavujú štyri funkcie: *schedule*, *start*, *scheduledstart* a *stop*.

Funkcia *schedule* je dvojstavová a slúži pre naplánovanie spustenia definovaného trafficu. V prvom stave generuje formulár, do ktorého užívateľ zadáva čas a dátum plánovaného spustenia a zastavenia toku. V druhom stave prebieha validácia zadaných časov, ďalej sa testuje porušenie ochranných časových odstupov medzi požiadavkami a oprávnenie pre spúšťanie traffic-ov na zvolenom prvku. Ak sú všetky podmienky splnené, aktualizuje stav trafficu v databáze a generuje riadky pre spustenie a zastavenie toku pre CRON. Riadky predstavujú otvorenie spúšťacej webovej stránky, kde sú parametre predané metódou GET. Riadky pre spustenie majú tvar výpisu 2, riadky pre zastavenie majú tvar výpisu 3.

```
wget https://oslo.dvirtlab.net/index.php?page=41
\&action=scheduedstart\&user=prd001\&secret=heslo\&id=119 --no-check-
certificate -O /dev/null  > /dev/null 2>&1 &
```

Výpis 2

```
wget https://oslo.dvirtlab.net/index.php?page=41\&action=stop
\&user=prd001\&secret=heslo\&id=114 --no-check-certificate -O
/dev/null  > /dev/null 2>&1 &
```

Výpis 3

Parameter *action* predstavuje volanú funkciu, *user* špecifikuje užívateľa, ktorý naplánoval tento traffic, *secret* predstavuje preddefinovaný reťazec zastupujúci heslo pre komunikáciu služby CRON so systémom a parameter *id* identifikuje naplánovaný tok. Pravosť certifikátu pre doménu nie je potrebné overovať a všetky výstupy budú zahadzované. Keďže služba CRON nepodporuje plánovanie s presnosťou na sekundu, riadky sú rozšírené ešte príkazom sleep. Riadky sú ukladané do súboru *cronfile* v hlavnom adresári Virtlab, odkiaľ sú vyberané a inštalované službou CRON.

Funkcia *start* predstavuje funkciu pre okamžité spustenie požadovaného trafficu. Po jej spustení funkcia kontroluje práva a bezpečné odstupy ako predchádzajúca. Po kontrole uzamyká prvok a podľa konfiguračného súboru generuje príkaz spustiteľný na prvku. Následne sa pripája na konzolu prvku pomocou ďalšieho konfiguračného súboru a odosiela príkaz na konzolu zariadenia. Všetky konfiguračné súbory sú popísane v nasledujúcej kapitole. Po vykonaní príkazu, a teda po spustení požadovaného trafficu aktualizuje jeho stav v databáze a tým odomyká prvok.

Funkcia *scheduledstart* predstavuje jedinú funkciu, ktorá nie je volaná priamo užívateľom. Ako je možné pozorovať z výpisu 2, je volaná službou CRON. Vzhľadom k tejto skutočnosti pre ňu platia špeciálne podmienky pre autentizáciu užívateľa. Užívateľ teda nemusí byť prihlásený v systéme a systém nekontroluje súhlas jeho session_id, ale sú kontrolované práva a tajný reťazec predaný funkcii metódou GET. Spúšťanie potom prebieha obdobne od pri funkcii *start*.

Poslednou funkciou tohto modulu je funkcia *stop*. Jej úloha spočíva v resetovaní stavu trafficu do pôvodného stavu. Teda ide o zrušenie rezervácie a prípadné zastavenie práve bežiaceho toku. Autentizácia užívateľa tu môže prebiehať dvoma spôsobmi a to buď štandardným spôsobom ak je volaná užívateľom alebo pomocou tajného reťazca ak je volaná službou CRON. Jej funkčnosť je obdobná, ako u predchádzajúcich funkcií.

4.2.4 traffic_multi.php

Modul je zodpovedný za prácu so zloženými tokmi. Jeho obsahom sú funkcie *create, delete, copy, addtraffic, deletemulti, schedule*.

Funkcia *edit* predstavuje funkciu z väčšej časti využívanú ostatnými. Jej zmyslom je poskytovanie rozhrania pre editáciu zloženého toku a ukladanie zmien. Základné operácie pri editácii sú pridanie elementárneho trafficu, odstránenie elementárneho trafficu, pomenovanie zloženého trafficu a nastavovanie časových odoziev pre spúšťanie a zastavovanie elementárnych trafficov.

Funkcia *create* predstavuje vytvorenie nového zloženého trafficu a následné volanie funkcie *edit* na novovytvorený zložený tok.

Funkcie *addtraffic, delete* a *deletemulti* predstavujú operácie pridania a odobratia elementárneho trafficu do/zo zloženého, prípadne následné odstránenie zloženého trafficu. Ide o prosté pridávanie a vymazávanie záznamov v databáze.

Náročnou funkciou je *copy*, vykonávajúcou kopírovanie zložených tokov. Pritom je potrebné skopírovať všetky elementárne toky s ich parametrami, skopírovať zložený tok a presmerovať všetky parametre novovytvoreného zloženého toku na skopírované elementárne toky. Opäť sa volá funkcia *edit* umožňujúca užívateľom pomenovať novovytvorený tok. Funkciu *copy* je možné použiť s parametrom public predávanú metódou GET. V tomto prípade ide o kopírovanie trafficov medzi preddefinované. Rovnako je tomu i v prípade funkcie *copy* v module traffic_generator.php.

Poslednou funkciou je funkcia *schedule*. Táto funkcia má dva stavy. V prvom je užívateľ vyzvaný formulárom pre zadanie počiatočného času, od ktorého sa začínajú počítať časové odozvy elementárnych trafficov. Systém následne prepočíta nové časy spúšťania a zastavovania elementárnych trafficov a odosiela ich funkcii *schedule* z modulu traffic_starter.php postupne pre všetky elementárne toky.

4.2.5 users-delete.php

Jedná sa o doplnenie pôvodného modulu pre vymazávanie užívateľov. Spolu s vymazaným užívateľom sa mažú i jeho privátne trafficy. Preddefinované trafficy sa nemažú.

4.2.6 tasks-delete.php

Jedná sa o doplnenie pôvodného modulu pre vymazávanie úloh. Spolu s úlohou sa mažú i všetky definované trafficy pre túto úlohu.

4.2.7 toggle.js

Modul obsahuje jedinú funkciu *toggle(id),* ktorej úlohou je zobrazovať a skrývať obsah tagu , ktorého id je rovné parametru predaného funkcii. Modul využívajú moduly traffic_generator.php a traffic_multi.php pre skrývanie častí formulára, ktoré nemajú byť viditeľné. Príkladom je skrývanie pokročilých nastavení pri editácii trafficu.

4.2.8 gen_validatorv31.js

Modul predstavuje validátor formulárov. Jednotlivým poliam formulára sú definované dátové typy. Pri pokuse o odoslanie formulára sú kontrolované požiadavky na dátové typy. Konfigurácia dátových typov pre polia je obsiahnutá v konfiguračných súboroch pre trafficy, popísaných v nasledujúcej kapitole. Validátor pozná nasledujúce dátové typy:

Dátový typ	Popis
req	Pole nesmie ostať prázdne
maxlen=???	Maximálna dĺžka poľa
minlen=???	Minimálna dĺžka poľa
alnum	Iba alfanumerické hodnoty
alnum_s	Alfanumerické hodnoty a medzery
num	Iba numerické hodnoty
dec	Iba desatinné čísla
alpha	Iba abecedné dáta
alpha_s	Abecedné dáta a medzery
email	Pole v tvare emailu
regexp=???	Pole spĺňuje regulárny výraz

4.2.9 consConnector.php.inc

Tento modul je reprezentovaný ako trieda. Jednotlivé inštancie predstavujú pripojenia na fyzickú konzolu zariadení. Trieda bola inšpirovaná modulom systému Virtlab pre testovanie splnenia úloh študentami.

Po vytvorení inštancie je potrebná inicializácia spojenia. Tú zabezpečuje metóda *connectToDevice($device, $idReservation).* Metóda obsahuje dva parametre, ktoré sú potrebné konzolovým serverom systému Virtlab pre vytvorenie spojenia. Prvým je identifikátor zariadenia *$device,* druhým je id rezervácie *$idReservation.* Metóda kontaktuje server a vyžiada vytvorenie pripojenia na konzolu zariadenia v exkluzívnom tutor móde, to znamená, že na konzolu zariadenia nebude môcť pristupovať nikto iný, pokiaľ nebude spojenie ukončené.

Ďalšími metódami sú *clearStream()* pre zahodenie znakov prichádzajúcich z konzoly, *sendCommand($command)* pre odoslanie textu obsiahnutého v parametre na konzolu zariadenia, *readReply()* pre prečítanie znakov prichádzajúcich z konzoly a *closeConnection()* pre odpojenie od konzoly zariadenia.

Táto trieda je využívaná modulom traffic_starter.php pre spúšťanie a zastavovanie generátorov trafficov. A taktiež modulom applet.php, pre odoslanie príkazu pre zobrazenie štatistík generátorov trafficu.

4.2.10 applet.php

Jedná sa o doplnenie pôvodného modulu, ktorý slúži pre umožnenie užívateľovi pristúpiť na konzolu zariadenia, pomocou klientského appletu. Pred spustením samotného appletu je na konzolu zariadenia odoslaná požiadavka pre zobrazenie štatistických dát. Tie si potom užívateľ môže prezrieť v okne appletu. Prípadne ak sa Štatistiky zobrazujú prostredníctvom interaktívnej aplikácie, má možnosť ju ovládať.

4.3 Analýza dátových tokov

Analyzovať toky na sieti je možné pomocou rôznych sieťových analyzátorov. Takéto analyzátory môžeme rozdeliť podľa schopnosti analyzovania na univerzálne a špeciálne.

Špeciálne analyzátory dokážu analyzovať iba určité typy toku, napríklad dokážu analyzovať telefonické hovory VoIP telefónie. Najčastejšie sú súčasťou aplikácií, ktoré takéto toky generujú. Pre systém Virtlab sú dôležité predovšetkým analyzátory s výstupom na textovú konzolu. Preto boli vybrané aplikácie obsahujúce ako svoju súčasť nástroje pre zobrazovanie štatistík prenesených dát. Generátory môžu buď bežať neustále na pozadí, prípadne sa spúšťať spolu s analyzovaným tokom. Teda analyzátory sú spúšťané rovnakým spôsobom ako aplikácie generujúce toky na sieti a ich spúšťacie parametre môžu byť pripojené v konfiguračných súboroch pre vytváranie trafficov. Výstupy analyzátorov je vhodné ukladať do jedného súboru a v prípade

ich vyžiadania ich poskytnúť užívateľovi. Užívateľ takto získa prehľad o tokoch v chronologickom poradí v ako sa na sieti vyskytovali.

Univerzálne analyzátory predstavujú aplikácie, ktoré zachytávajú všetky druhy trafficu na sieti. Ich nevýhodou je, že nedokážu popísať toky tak detailne ako analyzátory špeciálne. Sú však často schopné samostatne rozoznávať o aký druh toku na sieti sa jedná a poskytujú ucelenejší prehľad o stave siete. Výhodné sú i interaktívne aplikácie umožňujúce užívateľovi definovať vlastné filtre. Predstaviteľom takéhoto analyzátora je IPTraf.

Systém je navrhnutý tak, že umožňuje použiť ľubovoľný analyzátor, ktorý posiela svoj výstup na textovú konzolu. Po stlačení tlačidla pre zobrazenie výsledkov na prvku topológie je vykonaný príkaz definovaný v konfigurácii. Tento príkaz slúži na spustenie interaktívnych analyzátorov, prípadne na vypísanie nazbieraných dát na konzolu zariadenia. Následne je užívateľovi prezentovaný prístup na konzolu prostredníctvom klientského appletu, kde si užívateľ môže prezrieť výsledky, prípadne ovládať interaktívny analyzátor. Pri jednoduchom zobrazovaní výsledku treba počítať s odozvou spustenia appletu. Najlepšie zavedením príkazu pre čakanie na stlačenie klávesy. Bližšia konfigurácia je uvedená v nasledujúcej kapitole.

5 Inštalácia a konfigurácia

Pred možnosťou používania systému je potreba vytvoriť potrebné dátové štruktúry, teda vytvoriť odpovedajúce tabuľky v SRBD, nakopírovať do filesystému systému Virtlab súbory potrebné pre beh aplikácie a vykonať úvodnú konfiguráciu systému.

5.1 Vytvorenie tabuliek databázy

Prvý krok predstavuje vytvorenie odpovedajúcich tabuliek v SRBD. Pre vytvorenie je potrebné vykonať nasledujúce dopyty:

```
CREATE TABLE `traffic` (
  `traffic_id` int(11) unsigned NOT NULL auto_increment,
  `creator` varchar(20) collate utf8_czech_ci NOT NULL,
  `task_id` int(11) NOT NULL,
  `name` varchar(30) collate utf8_czech_ci NOT NULL,
  `config` text collate utf8_czech_ci NOT NULL COMMENT 'config file
     for this traffic',
  `vertex` varchar(150) collate utf8_czech_ci NOT NULL,
  `private` tinyint(1) NOT NULL,
  PRIMARY KEY  (`traffic_id`),
  KEY `creator` (`creator`),
  KEY `task_id` (`task_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;

CREATE TABLE `traffic_multischedule` (
  `id` int(11) NOT NULL auto_increment,
  `creator` varchar(20) collate utf8_czech_ci NOT NULL,
  `task_id` int(11) NOT NULL,
  `name` varchar(30) collate utf8_czech_ci NOT NULL,
  `private` tinyint(1) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;

CREATE TABLE `traffic_multischedule_parameters` (
  `multischedule_id` int(11) NOT NULL,
  `traffic_id` int(11) NOT NULL,
  `delay` int(11) NOT NULL,
  `delay_end` int(11) NOT NULL,
  KEY `multischedule_id` (`multischedule_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;

CREATE TABLE `traffic_parameters` (
  `traffic_id` int(11) NOT NULL COMMENT 'forein key form traffic',
  `advanced` tinyint(1) NOT NULL,
  `index` smallint(6) NOT NULL COMMENT 'parameter index from php
     config',
  `value` text collate utf8_czech_ci NOT NULL,
  KEY `traffic_id` (`traffic_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;
```

```
CREATE TABLE `traffic_schedule` (
  `id` bigint(20) NOT NULL auto_increment,
  `creator` varchar(20) collate utf8_czech_ci NOT NULL,
  `traffic_id` int(11) NOT NULL,
  `res_id` int(11) NOT NULL,
  `from` datetime default NULL,
  `to` datetime default NULL,
  `running` tinyint(1) NOT NULL,
  `vertex` varchar(150) collate utf8_czech_ci NOT NULL,
  `additional` text collate utf8_czech_ci,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;
```

Výpis 4

5.2 Vytvorenie / nakopírovanie súborov

Pre funkčnosť systému je potrebné nakopírovať, prípadne zlúčiť s existujúcimi nasledovné súbory:

virtlab/web/

- applet.php
- index.php

virtlab/web/virtlab/

- reser-active.php
- traffic_generator.php
- traffic_starter.php
- traffic_multi.php
- users-delete.php
- tasks-delete.php

virtlab/class/

- consConnector.php.inc
- virtlabLanguage.php.inc
- virtlabWeb.php.inc

virtlab/include/javascript

- toggle.js
- gen_validatorv31.js

/virtlab/img/traffic_icons

- *.png

- readme

5.3 Konfigurácia systému

Konfiguráciu systému predstavuje nastavenie niektorých systémových premenných a vytvorenie konfiguračných súborov pre spúšťanie generátorov trafficu. V nasledujúcej časti bude vysvetlený význam premenných a kde je potrebné ich nastaviť.

Konfigurácia začína v hlavnom konfiguračnom súbore webu Vitrlab: *virtlab/web/settings.php*. Tu je potreba zadefinovať nový formát pre zadávanie a zobrazovanie času a dátumu, keďže predchádzajúca implementácia nepracovala s presnosťou na sekundy. Pri práci s generátormi je však takáto presnosť potrebná. Definujeme nové premenné *$datesecformat* a *$datesecformat_sql*.

```
//date format - arg for date()
    $datesecformat    = '%d.%m.%Y %H:%M:%S';
    $datesecformat_sql= '%Y-%m-%d %H:%M:%S';
```

Výpis 5

Ďalej definujeme ochranné časové odstupy medzi požiadavkami na konzolu prvku topológie a tajný reťazec, ktorý je používaný pre komunikáciu systému z CRONom. Pri voľbe ochranného času je treba zohľadniť maximálny čas, za ktorý sa systému podarí spustiť traffic na prvku.

```
//Ochranny cas medzi schedulovanymi ulohami a spustanymi ulohami
    $protect = 10;
//Ochranny cas, minimalny casovy rozdiel medzi scheduled casmi, musi
byt vacsi ako $protect
    $scheduleprotect = 15;
//Tajne heslo pre komunikaciu z cronom, moze byt nahodne, symetricky
sa kopiruje na cron
    $secret = "tajneheslo";
```

Výpis 6

Konfigurácia pokračuje vytvorením konfiguračných súborov pre zostavenie príkazu pre spustenie generátoru toku s požadovanými. Tieto súbory sa nachádzajú v zložke *virtlab/web/virtlab/traffic_generator_configs*. Každý súbor v tomto adresári je schopný vygenerovať jeden príkaz pre generátor trafficu. Pomenovanie súboru má tvar *xxx_yyyyy.php*, kde xxx predstavuje trojpísmenovú skratku komunikačného jazyku konfiguračného súboru, napríklad *cze* a yyyyy predstavuje vhodné pomenovnie konfiguračného systému, pre rýchle odlíšenie v prípade editácie, napríklad *web-server*. Parametre konfigurácie budú vysvetlené na príklade pre traffic typu PING z nasledujúceho Výpisu.

```
<?php
//Konfiguracny subor generatoru traficu Virtlab pre PING
```

```
//Aky XML atribut os musi mat prvok pre spustenie tohoto trafficu
$generator_os[] = "";
$generator_os[] = "linux";

//Nazov polozky generatora
$generator_title = "PING";

//Cesta k spustitelnemu suboru
$generator_pathsender = "/bin/ping";

//Parametre aplikacie v poradi: realny parameter aplikacie, nazov
parametra vo formulari, typ parametra, predvyplnena hodnota
$generator_parameters[] = array(" ","Cílová ip adresa:","regexp=^([0-
2]{0,1}[0-9]{0,1}[0-9]{1}\.){3}[0-2]{0,1}[0-9]{0,1}[0-
9]{1}$","127.0.0.1");

//Parametre aplikacie nepotrebne pre spravnu funkciu (pokrocile vol-
by): realny parameter aplikacie, nazov parametra vo formulari, typ
parametra, predvyplnena hodnota
$generator_advancedparameters[] = array("-i ","Interval opaková-
ní:","dec","1");

//Dalsie parametre, ktore sa dopisuju vzdy
$generator_always = "";
?>
```

Výpis 7

$generator_os špecifikuje, že tento konfiguračný súbor bude použitý na všetkých prvkoch topológie, ktorých atribút *os* v XML pre popis topológie (viď kapitola 1.1) nadobúda niektorú z hodnôt atribútu. Teda že na zariadení s týmto atribútom os je možné generovať takýto typ trafficu. *$generator_title* predstavuje pomenovanie typu trafficu, ktoré sa bude zobrazovať užívateľom pri výbere typu trafficu.

Najdôležitejšiu časť konfiguračného súboru predstavuje *pole $generator_parameters,* resp. *$generator_advancedparameters.* V každom riadku tohoto poľa sa nachádzajú štyri prvky:

1. reálny parameter aplikácie, tj. prepínač príkazového riadku, napríklad –*i* pre PING
2. text popisujúci tento parameter, ktorý sa zobrazuje užívateľovi pri konfigurácii tohto parametru, napríklad *Interval opakovania* pre parameter –i pre PING
3. akceptovaný dátový typ, ktorý sa validuje pri zadávaní, napr. *dec* desatinné čislo, *alpha* pre pre abecedné dáta, alebo *regexp*=xxxx pre použitie regulárneho výrazu
4. predvyplnená hodnota

Systém následne spája reťazec uvedený v *$generator_pathsender* s vyplnenými parametrami od užívateľa a reťazcom uvedeným v *$generator_always*. Tým vytvára príkaz pre spustenie generátoru na prvku topológie. V prípade, že generátor má i vlastný analyzátor, je vhodné ho taktiež zahrnúť do spúšťacieho reťazca a presmerovať výstupné dáta na vhodné miesto.

Posledné miesto pre konfiguráciu systému je adresár *virtlab/web/virtlab/traffic_generator_configs/os_configs*, kde sa nachádzajú konfiguračné súbory špecifikujúce postup spustenia príkazu generovaného predchádzajúcim konfiguračným súborom. Každý konfiguračný súbor je pomenovaný *xxxx.php*, kde xxxx hovorí, že tento súbor bude použitý pre prvok topógie s XML atribútom *os* (viď kapitola 1.1) rovným xxxx. Prvky ktoré nemajú v XML špecifikovaný v atribút os sa budú teda riadiť súborom *.php*.

Každý z týchto súborov musí obsahovať implementáciu troch funkcií:

1. **starttraffic($connection,$command)**, kde parameter *$command* predstavuje príkaz, ktorý chceme spustiť na prvku a parameter *$connection* predstavuje inštanciu pripojenia na konzolu zariadenia vytvorenú systémom. Inštacia je vytvorená z triedy consConnector a sú pre ňu dostupné metódy:
   i. sendCommand($command) – odošle príkaz *$command* na konzolu zariadenia
   ii. clearStream() – zahodí všetky výpisy prichádzajúce z konzoly
   iii. readReply() – vracia výpis prichádzajúci z konzoly

   Funkcia vracia textový reťazec potrebný pre prípadné zastavenie, alebo prázdny reťazec ak došlo k chybe.

2. **endtraffic($connection,$text)**, kde parameter *$connection* je totožný so starttraffic() a parameter *$text* predstavuje text vrátený funkciou starttraffic(). Funkcia implementuje postup pre zastavenie trafficu na prvku.

3. **analyzator($connection)**, kde parameter *$connection* je totožný so starttraffic(). Funkcia implementuje postup pre zobrazenie štatistík na zariadení, ktoré je potom prezreté na konzole prostredníctvom klientského appletu.

Na nasledujúcom výpise je zobrazený príklad, ako môže vyzerať funkcia strattraffic() pre jednoduché spustenie príkazu na systéme Linux. Funkcia vracia PID spusteného procesu.

```
function starttraffic($connection,$command){
    $connection->sendCommand($command." > /dev/null 2>&1 &");
    $temp = $connection->readReply();
    $temp = preg_replace('#.*\[.+\]\s([0-9]+).*#si','\1',$temp);
    return $temp;
}
```

Výpis 8

Posledným krokom nastavenia je nastavenie služby CRON, ktorá bude spúšťať a zastavovať naplánované trafficy. Na systéme Linux predstavuje túto službu démon crond. Riadky pretohoto démona generuje systém do hlavného adresára systému Virtlab, do súboru cronfile. Je však potrebné, aby si systém plnil tabuľku démona z tohto súboru, hlavne z dôvodu, že server apache

nemá oprávnenia pre zápis do jeho tabuliek. Pod užívateľským účtom, z ktorého je možné posielať požiadavky a užívateľ má práva na CRON, vložíme do jeho tabuľky nasledujúci riadok:

```
* * * * * sleep 59; [ -s /opt/virtlab/cronfile ] && (((crontab -l;cat
/opt/virtlab/cronfile) | crontab -);cat /dev/null >
/opt/virtlab/cronfile)
```

Výpis 9

Démon crond bude každú minútu kontrolovať súbor cronfile v hlavnom adresári Virtlabu a zavedie riadky nájdené v ňom do svojej tabuľky. Tabuľa sa však rýchlo naplní a je treba ju vyprázdniť. Toto docielime pridaním riadka:

```
0 3 * * * cat /opt/virtlab/cronrefresh | crontab -
```

Výpis 10

Riadok odkazuje CRON na súbor cronrefresh, ktorý je potrebné vytvoriť a skopírovať do neho oba riadky. Plánovanie tejto úlohy je dané na dobu mimo prevádzkových hodín.

Záver

Cieľom práce bolo navrhnúť a implementovať systém pre generovanie reálnych sieťových tokov pre systém Virtlab. Užívateľ bol oboznámený so základnou architektúrou tohto systému.

Pre generovanie boli vybrané charakteristické toky, nachádzajúce sa predovšetkým v globálnej sieti Internet, s ohľadom na pomer výskytu týchto tokov v reálnych sieťach a úlohy, ktoré môžu študenti na systéme Virtlab vypracovávať. Kvôli požiadavke autentickosti tokov boli vybraté reálne aplikácie generujúce toky pri svojej bežnej činnosti. Vzhľadom na nemožnosť využitia všetkých druhov aplikácií boli ďalšie vhodné vybrané z kategórie generátorov – simulátorov.

Pre integráciu tokov do systému Virtlab, bol vytvorený užívateľsky konfigurovateľný systém predstavujúci rozhranie medzi reálnymi aplikáciami generátorov štandardným webovým rozhraním systému. Medzi hlavné atribúty patrí konfigurovateľnosť – do systému je možné pridať nový generátor a prispôsobiť systém vytvorením nového konfiguračného súboru a jednoduché ovládanie podobné ovládaniu skutočných aplikácií generujúcich odpovedajúce toky. Systém taktiež umožňuje vytvárané toky ukladať a tým zabezpečiť ich opakovateľnosť. Správca úlohy má možnosť preddefinovania tokov použiteľných ostatnými užívateľmi systému. Tiež umožňuje i plánovanie zložitých tokov zložených z rôznych druhov tokov.

V neposlednom rade je užívateľ oboznámený s možnosťami sledovania a vyhodnocovania tokov na sieti. Je predstavený spôsob zavedenia rôznych druhov analyzátorov tokov, ich konfigurácie a zobrazovanie výsledkov.

Návrh systému pamätá i na ďalšie možné rozširovanie systému. Systém preto nie je obmedzený iba na generovanie tokov na koncových zariadenia typu pc, ale taktiež na ostatných prvkoch topológie s touto funkčnosťou a nie je viazaný ani na operačný systém zariadení. Postačuje vytvoriť nový konfiguračný súbor špecifikujúci spôsob pripojenia na zariadenie.

Citovaná literatúra

[1.] *Virtlab Wiki* [online]. 27. 8. 2008 [cit. 2010-05-03]. Virtuální laboratoř počítačových sítí. Dostupné z WWW: <www.virtlab.cz>.

[2.] AVALLONE, Stefano; PESCAPÈ, Antonio. *D-ITG* [online]. August, 2004 [cit. 2010-05-04]. D-ITG, Distributed Internet Traffic Generator. Dostupné z WWW: <http://www.grid.unina.it/software/ITG/>

[3.] *SIPp* [online]. 2004 [cit. 2010-05-04]. Welcome to SIPp. Dostupné z WWW: <http://sipp.sourceforge.net/>

[4.] HOFFMAN, John. *BitTornado* [online]. ? [cit. 2010-05-04]. BitTornado. Dostupné z WWW: <http://www.bittornado.com/>

[5.] GERARD, Paul. *IPTraf* [online]. 1997 [cit. 2010-05-05]. IPTraf IP Network Monitoring Software. Dostupné z WWW: <http://iptraf.seul.org/>

[6.] EVANS, Chris. *Vsftpd* [online]. ? [cit. 2010-05-04]. Vsftpd. Dostupné z WWW: <http://vsftpd.beasts.org/>

[7.] *BitTorrent Wikipedia* [online]. 2002 [cit. 2010-05-04]. BitTorrent. Dostupné z WWW: <http://cs.wikipedia.org/wiki/BitTorrent>

[8.] *The Apache Software Foundation* [online]. 1993 [cit. 2010-05-06]. The Apache Software Foundation. Dostupné z WWW: <http://www.apache.org/>